هوش مصنوعی **پاسخنامه تمرین نوشتاری شماره ۱**

مدرس: **دکتر فدایی**

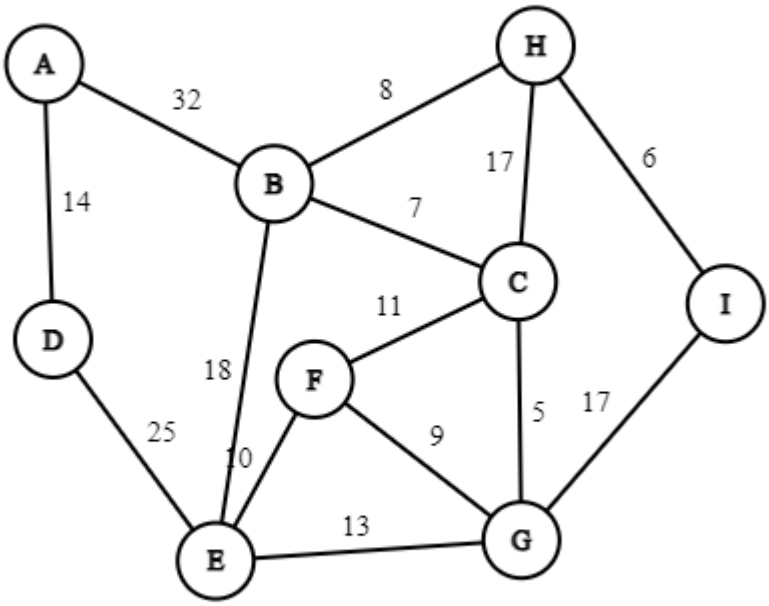
طراحان: امید پناکاری، پویا صادقی، علی اخگری، مجید فریدفر

مهلت تحویل: یکشنبه ۲۱ آبان ۱۴۰۲، ساعت ۲۳:۵۹

Search

سوال اول

در گراف زیر می‌خواهیم با کوتاهترین مسیر از راس A به I را بیابیم.



با استفاده از الگوریتم Uniform Cost Search کوتاهترین مسیر از A به I را بیابید. در صورت وجود چندین گزینه در یک مرحله، راس کوچکتر از نظر الفبایی را انتخاب کنید. در طی تمام مراحل مجموعه frontier، مجموعه explored و آخرین راس اضافه شده به explored در آن مرحله را به همراه مسیر طی شده و هزینه مصرف شده تا آن راس را بنویسید.

پاسخ

Frontier	Explored	Node	Cost	Path
B, D	A	A	0	A
B, E	A, D	D	14	A, D
E, H, C	A, D, B	B	32	A, B
E, H, G, F	A, D, B, C	C	39	A, B, C
H, G, F	A, D, B, C, E	E	39	A, D, E
G, F, I	A, D, B, C, E, H	H	40	A, B, H
F, I	A, D, B, C, E, H, G	G	44	A, B, C, G
F	A, D, B, C, E, H, G, I	I	46	A, B, H, I

سوال دوم

فرض کنید که یک درخت binary و n راسی داریم که در هر راس آن یک عدد از میان 1 تا n نوشته شده است. می‌خواهیم اعداد بر روی این درخت را به گونه‌ای جابه‌جا کنیم که درخت نهایی یک binary search tree باشد(اعداد واقع در زیر درخت سمت راست و چپ هر راس به ترتیب از عدد آن راس بزرگتر و کوچکتر باشند). در هر مرحله می‌توانیم اعداد دو راس مجاور را با یکدیگر جابه‌جا کنیم. برای حل این مساله یک heuristic ارائه دهید و admissible و consistent بودن آن را اثبات کنید.

پاسخ

اگر پس از تعدادی حرکت، درخت به درخت binary search تبدیل شود، هر عدد در جای مشخصی قرار خواهد گرفت. فرض کنیم برای هر عدد i دو راس s_i و t_i به ترتیب مکان آن عدد در درخت کنونی و درخت binary search نهایی باشند و همچنین $d(u,v)$ برابر فاصله دو راس u و v بر روی درخت باشد. آن‌گاه heuristic خود را برای این مساله به صورت زیر تعریف می‌کنیم:

$$H(T) = \frac{1}{2}\sum_{i=1}^{n} d(s_i, t_i)$$

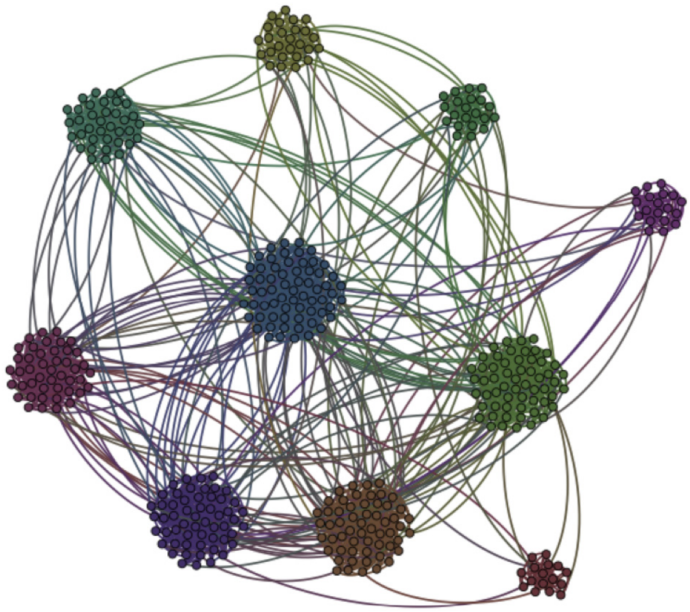
حال دو ویژگی consistency و admissiblity را برای این heuristic اثبات می‌کنیم.

می‌دانیم برای آن که درخت T به binary search tree تبدیل شود، باید عدد i حداقل به اندازه $d(s_i, t_i)$ بر روی درخت جابه‌جا شود و زمانی به درخت binary search رسیده‌ایم که مقدار عبارت $\sum_{i=1}^{n} d(s_i, t_i)$ برابر ۰ شود. همچنین در هر حرکت دو عدد مجاور بر روی درخت با یکدیگر جابه‌جا می‌شوند که به این معنی است که مقدار $\sum_{i=1}^{n} d(s_i, t_i)$ حداکثر دو واحد تغییر خواهد کرد. بنابراین برای ۰ کردن مقدار عبارت گفته شده به حداقل $\frac{1}{2}\sum_{i=1}^{n} d(s_i, t_i)$ حرکت نیاز می‌باشد. بنابراین heuristic داده شده admissible می‌باشد.

همچنین اگر پس از k حرکت از درخت S به T برسیم، مقدار $\sum_{i=1}^{n} d(s_i, t_i)$ حداکثر $2k$ واحد تغییر می کند که به این معنی است که $H(T) - H(S) \leq k$. نتیجه می‌گیریم که heuristic داده شده consistent نیز می‌باشد.

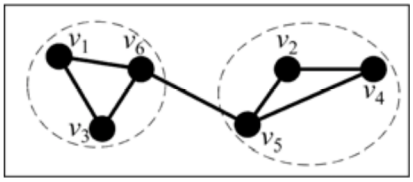
Genetic

در این قسمت، با استفاده از الگوریتم ژنتیک، به دنبال یافتن یک پاسخ خوب برای یک مسئله هستیم. در این مسئله، هدف ما پیدا کردن گره‌هایی در یک گراف است که درون گروه‌های خود ارتباطات قوی‌تری دارند. به طور دقیق‌تر، ما به دنبال شناسایی گروه‌هایی از این گره‌ها هستیم که بین اعضای گروه، ارتباطات چگال‌تری وجود دارد. توجه داشته باشید که در این مسئله، هر گره فقط به یک گروه تعلق خواهد داشت. به عبارت دیگر امکان اینکه یک گره به بیش از دو گروه تعلق داشته باشد وجود ندارد. همچنین امکان اینکه یک گره به هیچ گروهی تعلق نداشته باشد نیز امکان پذیر نیست.



الف) ژن و کروموزم‌های این مسئله را چگونه تعریف می‌کنید؟

ب) در این مسئله، شکل زیر نشان دهنده یک شبکه شامل ۶ گره و ۲ گروه می‌باشد. این شکل را در مسئله چگونه مشخص می‌کنید؟ (به عبارت دیگر یک encoding از شکل زیر ارائه دهید که بتوانید از آن به عنوان کروموزم در مسئله ژنتیک خود استفاده کنید)



ج) میوتیشن را چگونه تعریف می‌کنید؟ آیا سناریویی وجود دارد که ژن‌های حاصل از میوتیشن نیاز به بازنگری داشته باشند؟ (توجه کنید که میوتیشن باید براساس مدلی باشد که در بخش ب تعریف کرده‌اید)
د) کراس‌اور را چگونه تعریف می‌کنید؟ آیا سناریویی وجود دارد که ژن‌های حاصل از کراس‌اور نیاز به بازنگری داشته باشند؟ (توجه کنید که کراس‌اور باید براساس مدلی باشد که در بخش ب تعریف کرده‌اید)
ه) یک fitness function برای ارزیابی خوب بودن این مسئله ارائه دهید. (امتیازی)

پاسخ

الف) در این مسئله، ژن‌ها نماینده‌ی گره‌ها هستند و نشان می‌دهند که هر گره به کدام اجتماع تعلق دارد. همچنین کروموزم‌ها از چند ژن تشکیل شده‌اند که نماینده‌ی یک جواب احتمالی می‌باشند.
ب) یک روش encoding می‌تواند ماتریس زیر باشد:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

در این ماتریس تعداد ستون‌ها نشان‌دهنده تعداد گروه‌ها و تعداد سطرها نشان‌دهنده تعداد گره‌ها می‌باشد. همچنین یک یا صفر بودن به معنای تعلق داشتن یا نداشتن یک گره به آن گروه است.
ج) راه‌های متفاوتی برای میوتیشن می‌توان در نظر گرفت. ساده ترین کار این است که به احتمال mutation rate، یک راس را از یک گروه خارج و به گروه دیگری ملحق کنیم.
سناریویی که در آن لازم است ژن‌های حاصل از میوتیشن را بازنگری کنیم، می‌تواند شکل زیر باشد، زیرا نمی‌تواند یک گره به هر دو گروه تعلق داشته باشد یا به گروهی تعلق نداشته باشد.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

د) راه‌های متفاوتی برای کراس‌اور می‌توان در نظر گرفت. ساده ترین کار این است که به احتمال crossover rate، از روش k-point استفاده کنیم.
سناریویی که در آن لازم است ژن‌های حاصل از کراس‌اور را بازنگری کنیم، می‌تواند مانند قسمت ج باشد.
ه) می‌توان از معیار modularity استفاده کرد. به طور مشخص‌تر از این فرمول:

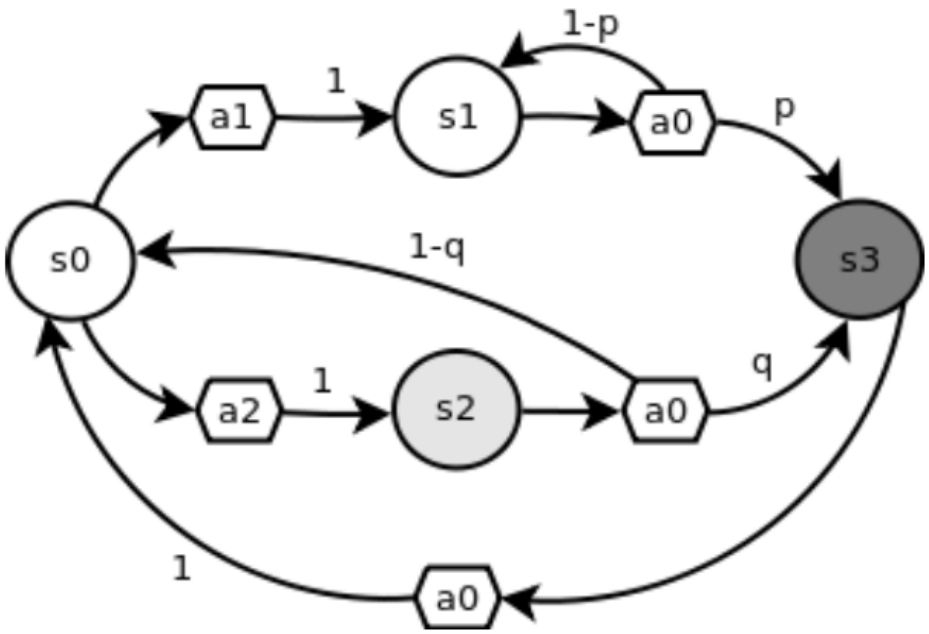
$$Q = \frac{1}{2m} \sum_{i,j} [a_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j)$$

در این فرمول، m تعداد یال‌ها می‌باشد. a_{ij} نشان‌دهنده که آیا دو رأس i و j به هم متصل هستند یا خیر، اگر متصل باشند $a_{ij} = 1$ و در غیر اینصورت، $a_{ij} = 0$ می‌باشد. c_i و c_j نشان دهنده گروه‌هایی است که رئوس i و j به آن تعلق دارند. به عبارت دیگر رأس i به c_i و رأس j به c_j تعلق دارد. اگر $c_i = c_j$ باشد، $\delta(c_i, c_j) = 1$ و در غیر اینصورت صفر خواهد بود. k_i و k_j نشان‌دهنده درجه رأس های i و j می‌باشند. توجه داشته باشید که $|Q| \leq 1$ می‌باشد و هر چه این مقدار به یک نزدیک تر باشد، نشان‌دهنده یک ارتباط چگال‌تر بین اعضای گروه می‌باشد.

MDP

سوال اول

تصویر زیر بیانگر مسئله MDP افق بینهایت m و با پارامترهای نرخ تخفیف $\gamma \in [0, 1)$ می‌باشد. در این تصویر، state ها با دایره و action ها با شش ضلعی نمایش داده شده اند. عدد نمایش داده شده برروی یال‌های جهت دار، بیانگر احتمال آن انتقال (transition probability) می‌باشد؛ به عنوان مثال $P(s3|s2, a0) = q$. یال‌های نمایش داده نشده، بیانگر احتمال صفر می‌باشند؛ به عنوان مثال $P(s0|s0, a0) = 0$. پاداش در رسیدن به وضعیت s3 برابر با ۱۰ و در رسیدن به وضعیت s2 برابر با ۱ می‌باشد و در غیر این دو وضعیت پاداش صفر است. همچنین $p, q \in [0, 1]$. با توجه به این موارد به سوالات مطرح شده پاسخ بدهید.



- تمام پالیسی های ممکن برای m را ذکر کنید.
- معادله بیانگر value function بهینه برای هر وضعیت را بنویسید. $(V^*(s_0), V^*(s_1), V^*(s_2), V^*(s_3))$
- آیا هیچ مقداری برای پارامتر p وجود دارد که به ازای تمام $q \in [0, 1]$ و $\gamma \in [0, 1)$، داشته باشیم $\pi^*(s_0) = a2$ ؟ توضیح دهید.
- آیا هیچ مقداری برای پارامتر q وجود دارد که به ازای $p > 0$ و $\gamma \in [0, 1)$، داشته باشیم $\pi^*(s_0) = a1$ ؟ توضیح دهید.
- با استفاده از $\gamma = 0.9$ و $p = q = 0.25$، V^* و π^* را برای تمامی state ها محاسبه کنید. میتوانید از معادلات بخش دوم و یا value iteration استفاده کنید. خطای $\varepsilon = 10^{-3}$ را بین V^* و V^t قابل پذیرش است.

پاسخ سوال اول

الف) تمام پالیسی های ممکن

دو پالیسی قابل تصور اند:

	S0	S1	S2	S3
π1	a1	a0	a0	a0
π2	a2	a0	a0	a0

ب) معادلات بیانگر value function بهینه:

$$V^*(s_0) = \max_{a\in\{a_1,a_2\}} 0+\gamma\sum_{s'} P(s'|s,a)V^*(s') = \gamma\max\{V^*(s_1), V^*(s_2)\}$$

$$V^*(s_1) = \max_{a\in\{a_0\}} 0+\gamma\sum_{s'} P(s'|s,a)V^*(s') = \gamma[(1-p)V^*(s_1)+pV^*(s_3)]$$

$$V^*(s_2) = \max_{a\in\{a_0\}} 1+\gamma\sum_{s'} P(s'|s,a)V^*(s') = 1+\gamma[(1-q)V^*(s_0)+qV^*(s_3)]$$

$$V^*(s_3) = \max_{a\in\{a_0\}} 10+\gamma\sum_{s'} P(s'|s,a)V^*(s') = 10+\gamma V^*(s_0)$$

ج)

توجه شود که $\pi*(s0) = argmax_{a\in\{a1,a2\}} \gamma Ps' P(s'|s,a)V*(s')$ که اگر قرار دهیم $\gamma = 0$ آنگاه a1 و a2 به یکدیگر بهینه خواهند خورد و لذا π^* بهینه نخواهد شد و در نتیجه آن ما نمیتوانیم تضمین کنیم که $\pi*(s0) = a2$. توجه شود که اگر $\gamma > 0$ آنگاه $\pi*(s0) = a2$ خواهد بود، اگر و تنها اگر $V*(s2) > V*(s1)$. اگر $p = 0$ آنگاه $V*(s1) = \gamma V*(s1) = 0$ و از آنجا که $V*(s2) = 1 + \gamma[(1-q)V*(s0) + qV*(s3)]\geq 1$ آنگاه خواهیم داشت که $V*(s2) > V*(s1)$ و $\pi*(s0) = a2$ به ازای هر $q\in[0,1$ و $\gamma\in(0,1)$.

د)

خیر. از آنجا که $V*(s2) = 1 + \gamma[(1-q)V*(s0) + qV*(s3)]\geq 1$ داریم که $\pi*(s0) = a1$ اگر و تنها اگر $V*(s1) > V*(s2)\geq 1$ به ازای تمام $p > 0$ و $\gamma\in[0,1)$. داریم که $V*(s1) = \gamma[(1-p)V*(s1) + pV*(s3)] = \frac{\gamma pV*(s3)}{1-\gamma(1-p)}$ پس در نتیجه همواره میتوانیم یک γ به اندازه کافی کوچک انتخاب کنیم به طوری که $V*(s1) < 1$. همچنین اگر مقدار صفر برای آن انتخاب شود، همانند بخش قبل $\pi*$ یکتا نخواهد بود.

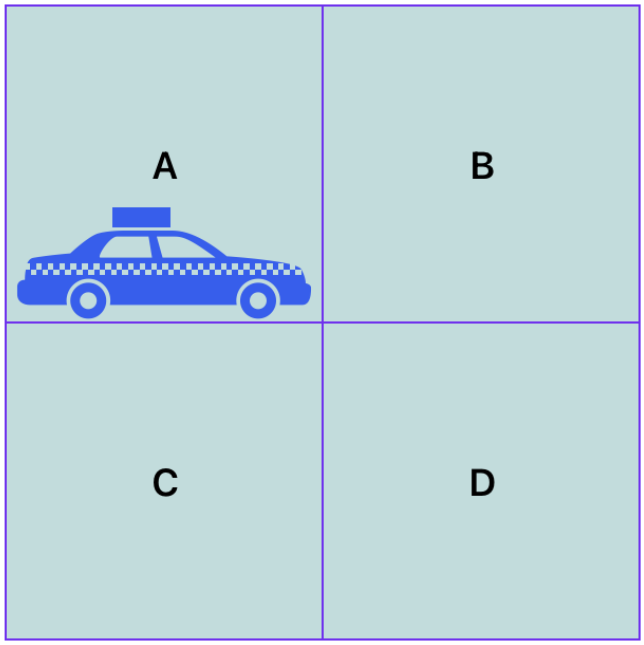
ه)

	S0	S1	S2	S3
V*	14.1846	15.7608	15.6969	22.7661
π*	a1	a0	a0	a0

سوال دوم

موقعیت زیر را درنظر بگیرید:

- فرض کنید شما یک راننده تاکسی در شهری با چهار مکان A، B، C و D هستید. شما می توانید در هر مکانی مسافران را سوار و پیاده کنید. شما برای هر سفر موفق، بسته به فاصله مبدا و مقصد، مبلغ ثابتی کسب می کنید. شما همچنین برای هر مایلی که رانندگی می کنید، چه با مسافر یا بدون مسافر، هزینه ای متحمل می شوید. شما می توانید انتخاب کنید که در هر مکانی بمانید و منتظر مسافر باشید یا به مکان همسایه رانندگی کنید و در آنجا به دنبال مسافر بگردید. احتمال پیدا کردن مسافر در هر مکان متفاوت است و ممکن است در طول زمان تغییر کند. هدف شما این است که سود کل مورد انتظار خود را در یک روز کاری به حداکثر برسانید.



با مشخص کردن اجزای زیر، این مسئله را به عنوان یک MDP مطرح کنید:

1. فضای حالت: حالت های احتمالی که می توانید در آن باشید چیست؟
2. فضای عمل: اقدامات ممکنی که می توانید در هر وضعیت انجام دهید چیست؟
3. مدل انتقال: با توجه به یک عمل، احتمال انتقال از یک حالت به حالت دیگر چقدر است؟
4. تابع پاداش: پاداش (یا هزینه) فوری برای هر جفت حالت-عمل چقدر است؟

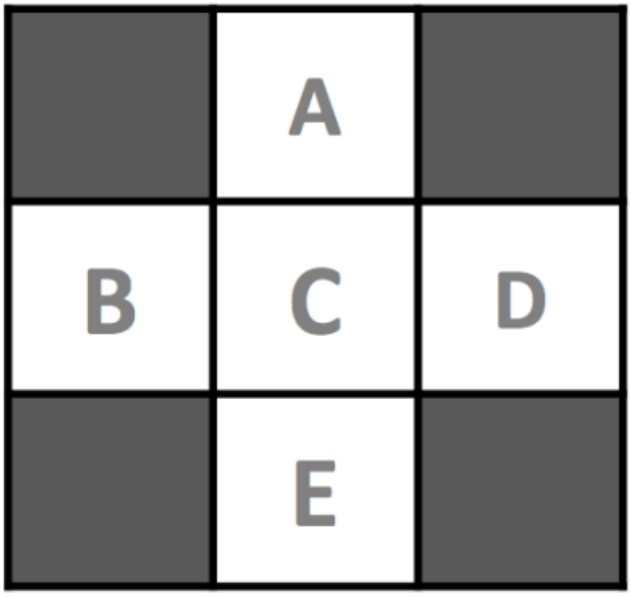
پاسخ سوال دوم

1. فضای حالت را می توان به عنوان ترکیبی از مکان فعلی شما و اینکه آیا مسافر دارید یا خیر تعریف کرد. به عنوان مثال، یک حالت ممکن (A، بله) است، به این معنی که شما در مکان A هستید و یک مسافر دارید. در مجموع هشت استیت ممکن وجود دارد.
2. فضای عمل را می توان به عنوان انتخاب ماندن در مکان فعلی یا حرکت به یک مکان همسایه تعریف کرد. به عنوان مثال، اگر در مکان A هستید، می توانید در آنجا بمانید یا به B یا C بروید. برای هر استیت سه اقدام ممکن وجود دارد.
3. مدل انتقال را می توان به عنوان احتمال یافتن مسافر در هر مکان، ضرب در احتمال مقصد مسافر تعریف کرد. به عنوان مثال، اگر در مکان A هستید و انتخاب می کنید که در آنجا بمانید، احتمال حرکت به حالت (A، بله) احتمال پیدا کردن مسافر در A است، برابر احتمال اینکه مسافر می خواهد به A برود (که صفر است). احتمال جابجایی به حالت (B، بله) احتمال یافتن مسافر در A است، ضربدر احتمال اینکه مسافر بخواهد به B برود و الی آخر.
4. تابع پاداش را می توان به عنوان تفاوت بین درآمد و هزینه برای هر جفت حالت-اقدام تعریف کرد. به عنوان مثال، اگر در استیت (A، بله) هستید و انتخاب می کنید به B بروید، پاداش، درآمدی است که برای پیاده کردن مسافر در B دریافت می کنید، منهای هزینه رانندگی از A به B. اگر استیت (A، خیر)

هستید و شما انتخاب می کنید که آنجا بمانید، پاداش صفر است، زیرا نه پولی کسب می کنید و نه خرج می کنید.

RL

فرض کنید یک ربات، در گریدورلد (Gridworld) زیر در حال جمع‌آوری اطلاعات است. او از یک استیت دلخواه شروع می‌کند و با انجام دادن اکشن‌های تصادفی، و با توجه به جایزه‌ای (reward) که به دست می‌آورد، سعی می‌کند به شناخت کافی از این محیط برسد و پالیسی بهینه را به دست بیاورد. هم‌چنین برای انجام این کار از الگوریتم Q-Learning که در درس با آن آشنا شده‌اید استفاده می‌کند.



توجه کنید که در این environment، پنج استیت (A, B, C, D, E) وجود دارد و در هر استیت ایجنت می‌تواند یکی از اکشن‌های West، East، Up و یا South را انجام دهد.

الف)

فرض کنید ایجنت چهار اکشن زیر را به صورت تصادفی در این محیط انجام می‌دهد:

1. از استیت B با انجام اکشن East به استیت C می‌رود، و 2 واحد جایزه می‌گیرد.
2. از استیت C با انجام اکشن South به استیت E می‌رود، و 4 واحد جایزه می‌گیرد.
3. از استیت C با انجام اکشن East به استیت A می‌رود، و 6 واحد جایزه می‌گیرد.
4. از استیت B با انجام اکشن East به استیت C می‌رود، و 2 واحد جایزه می‌گیرد.

بعد از انجام هر مرحله، Q-Table را رسم کنید (در ابتدا تمامی مقادیر جدول برابر صفر هستند) و نحوه‌ی آپدیت شدن جدول را نشان دهید. مقدار learning rate را برابر 0.5 و مقدار discount factor را برابر 1 در نظر بگیرید.

ب)

یک نسخه از الگوریتم Q-Learning با روش Epsilon-gready را در نظر بگیرید که به جای استفاده از پالیسی استخراج شده از Q-Table فعلی، از یک پالیسی ثابت استفاده می‌کنیم و با احتمال اپسیلون هنوز هم اکتشاف را انجام می دهیم. اگر این پالیسی ثابت بهینه باشد، عملکرد این الگوریتم چگونه با Q-Learning Epsilon-gready عادی مقایسه می شود؟

پاسخ

الف)

مقادیر نهایی Q-Table در خانه‌های (B, East) و (C, South) و (C, East) بعد از انجام هر اکشن در جدول زیر نشان داده شده است. واضح است که مقادیر بقیه‌ی خانه‌های جدول صفر باقی می‌مانند.

Transitions	$(B, East)$	$(C, South)$	$(C, East)$
(initial)	0	0	0
$(B, East, C, 2)$	1	0	0
$(C, South, E, 4)$	1	2	0
$(C, East, A, 6)$	1	2	3
$(B, East, C, 2)$	3	2	3

هم‌چنین آپدیت‌ها به این صورت شکل می‌گیرند:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + (\alpha)\left[r + \gamma \max_{a'} Q(s',a')\right]$$

1. Q(B, East) = (1 - 0.5)*0 + 0.5*(2 + 0) = 1
2. Q(C, South) = (1 - 0.5)*0 + 0.5*(4 + 0) = 2
3. Q(C, East) = (1 - 0.5)*0 + 0.5*(6 + 0) = 3
4. Q(B, East) = (1 - 0.5)*1 + 0.5*(2 + 3) = 3

ب)

هر دو الگوریتم در نهایت منجر به پیدا کردن Q-values بهینه خواهند شد. با این حال، Q-Learning Epsilon-gready عادی در طول راه بیشتر اشتباه می کند و باعث می شود که تفاوت بین پاداش واقعی و پاداش انتظاری بهینه بیشتر شود.

در عمل، Q-Learning Epsilon-gready عادی با اپسیلون کوچک ممکن است منجر به پالیسی‌ای شود که "خیلی خوب" است اما بهینه نیست، بنابراین بسیار بعید است که تغییر کند مگر اینکه تعداد بسیار زیادی تکرار داده شود تا فرصتی برای یافتن یک پالیسی بهتر به صورت تصادفی فراهم شود. این نتیجه به عنوان یک بیشینه محلی شناخته می شود.